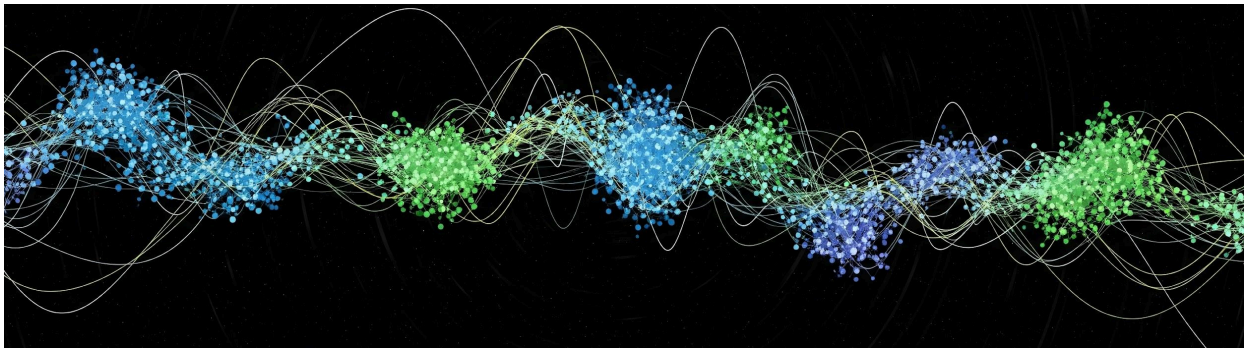

Regression Exercise: Insurance Premium Prediction

Objective

The goal of this exercise is to build a machine learning model that accurately predicts insurance premium amounts based on various policyholder and insurance-related factors. This is a regression task, where the target variable (Premium Amount) is continuous.

Dataset

The dataset used for this exercise is from the Kaggle Playground Series Season 4, Episode 12 competition: Regression with an Insurance Dataset.

The dataset consists of two main files:

- **train.csv**: Contains the training data, including the target variable (Premium Amount).
- **test.csv**: Contains the test data for which you need to predict the Premium Amount.
- **sample_submission.csv**: A sample submission file showing the required format for predictions.

Dataset Description

The dataset for this competition was generated from a deep learning model trained on an original insurance premium prediction dataset. It includes various

features that describe the policyholder and aspects related to their insurance. While specific feature names are not exhaustively listed here, common factors in insurance datasets often include:

- **Demographic information**: Age, gender, education level, marital status, occupation.
- **Health-related factors**: Health score, smoking status, exercise frequency.
- **Insurance policy details**: Property type, policy type, previous claims.
- **Geographical information**: Location or city-related codes.
- **Other relevant details**: Customer feedback, vehicle age (if applicable to vehicle insurance).
- **Premium Amount**: (Target Variable) The continuous numerical value representing the insurance premium.

Tasks

1. Data Loading and Exploration

- Load the train.csv and test.csv datasets using a data manipulation library (e.g., Pandas in Python).
- Perform an initial exploratory data analysis (EDA) to understand the data distribution, identify missing values (if any), and observe relationships between features and the target variable.
- Visualize key relationships and distributions (e.g., histograms, scatter plots, correlation matrix). Pay attention to the distribution of the Premium Amount.

2. Data Preprocessing

- Handle any missing values.
- Address categorical features. Consider techniques like one-hot encoding or label encoding.
- Scale numerical features if necessary (e.g., using StandardScaler or MinMaxScaler).
- **Important Consideration for Target Variable**: The evaluation metric for this competition is Root Mean Squared Log Error (RMSLE). This often suggests that applying a log transformation (e.g., np.log1p) to the Premium Amount might improve model performance, especially if your chosen model optimizes for Mean Squared Error (MSE) or Root Mean Squared Error

(RMSE). Remember to transform predictions back using np.expm1 before evaluating or submitting.
- Feature engineering: Consider creating new features from existing ones if you believe it will improve model performance.

3. Model Training

- Split the train.csv dataset into training and validation sets to evaluate your model's performance during development.
- Choose one or more regression algorithms (e.g., Linear Regression, Decision Tree Regressor).
- Train your chosen model(s) on the training data.

4. Model Evaluation

- Evaluate your model's performance on the validation set using the Root Mean Squared Log Error (RMSLE) metric. RMSLE is particularly useful when you want to penalize under-predictions more than over-predictions, and when the target variable has a wide range of values or a skewed distribution.
- Experiment with hyperparameter tuning to optimize your model's performance.

5. Prediction and Submission (Optional)

- Once you are satisfied with your model's performance on the validation set, make predictions on the test.csv dataset. Remember to inverse-transform your predictions if you applied a log transformation earlier.
- Format your predictions into a CSV file with `id` and `Premium Amount` columns, similar to `sample_submission.csv`. This step can be used to simulate a competition submission.

Deliverables

- A Python script or Jupyter Notebook (.ipynb file) containing your code for data loading, preprocessing, model training, and evaluation.
- A brief report summarizing your approach, key findings from EDA, model choices, challenges encountered, and the final RMSLE score on your

validation set. If you generated predictions on the test set, include insights from that as well.

Tips for Students

- Focus on understanding the evaluation metric (RMSLE) and how it influences your data preprocessing (especially for the target variable) and model selection.
- Explore different ways to handle categorical features, as they often play a crucial role in tabular datasets.
- Consider using cross-validation for more robust model evaluation.
- Refer to Kaggle notebooks and discussions from the competition for inspiration and to learn about common approaches, but ensure you understand the concepts and implement them yourself.